UDC 510.67:004.652

E. Malakhov, ScD.,
D. Shchelkonogov

STABILITY AND INEVITABILITY OF CONNECTIONS OF SUBJECT DOMAINS

Abstract. *In this paper a problem of the mathematical description of connections between objects is considered as one of the most fundamental elements of a model of a subject domain. Analysis of connections of an arbitrary subject domain is presented. Notions of stability and inevitability of connections are introduced and their quantitative estimates are given.*

Keywords: *subject domain modelling, connection stability, connection inevitability, logical connection, fraternal connection, mass problem*

Е. В. Малахов, д-р техн. наук,
Д. О. Щелконогов

СТАБИЛЬНОСТЬ И НЕИЗБЕЖНОСТЬ СВЯЗЕЙ ПРЕДМЕТНОЙ ОБЛАСТИ

Аннотация: *В статье рассматривается проблема математического описания связей между объектами, как одного из фундаментальных элементов модели предметной области. Представлен анализ связей в произвольной предметной области. Введены понятие стабильности и неизбежности связей и её количественные характеристики.*

Ключевые слова: *моделирование предметной области, стабильность связи, неизбежность связи, логическая связь, братская связь, массовая проблема*

Є. В. Малахов, д-р техн. наук,
Д. О. Щелконогов

СТАБІЛЬНІСТЬ ТА НЕВІДВОРОТНІСТЬ ЗВ'ЯЗКІВ ПРЕДМЕТНОЇ ОБЛАСТІ

Анотація: *У статті розглядається проблема математичного опису зв'язків між об'єктами, як одного з фундаментальних елементів моделі предметної області. Представлено аналіз зв'язків у довільній предметній області. Введені поняття стабільності та невідворотності зв'язків та її кількісні характеристики.*

Ключові слова: *моделювання предметної області, стабільність зв'язку, невідворотність зв'язку, логічній зв'язок, братський зв'язок, масова проблема*

Introduction. The goal of this article is to analyze connections between objects inside a subject domain.

Connections between objects are one of the three fundamental parts of the definition of a subject domain [1] beside objects and mass problems. Connections allow describing a structure of a subject domain, relations between objects; give a subject domain its own unique qualities. The same set of objects can describe different subject domains if those subject domains have different sets of connections in the same manner as the same set of atoms can describe different molecules depending on the inner structure of connections.

Connections between instances allow determining the state in which a subject domain exists in the given moment of time, and the qualities that it has in the given moment of time from the whole set of possible qualities defined by the connections between objects. Connections between instances of objects apart from connections between objects of a subject domain can be changed as a result of influences on the subject domain, transferring the subject domain to the new state. Wise change of such connections can result in the qualitative change of a state of a subject domain: make it better or worse. For example, if we take pupils and teachers as objects, then by connecting pupils to better teachers we can augment quality of the education in a subject domain "school" without changing any instances of objects of that subject domain.

Existing theories on connections. Connections between different objects have been widely studied in different areas of mathematics and computer sciences. Among the most significant results we should mention graph theory, theory of relational databases and object-oriented modelling.

Graph theory [2 – 3] as one of the most abstract theories on connections allows describing different relations between vertexes which can correspond to any entities in the real world and allow solving the most abstract problems, such as shortest paths search, and others. Nevertheless, given that enormous power, instruments of that theory don't fully support possibility of working with semantics of a subject domain and problems that are solved on it.

The theory of relational databases [4] and data warehouses [5] allow describing connections between objects of a subject domain (relations) and instances of those objects (entities). Although those connections are taken into account while solving problems, as in the graph theory the reason why they are used is solely technical. They are not used to improve quality of the tasks being solved; they used just to create a model of a subject domain. The classification of connections is given here (1-to-1, 1-to-many, many-to-1, and many-to-many). It needs to be mentioned, that the connection many-to-many is implemented by the auxiliary (artificial) "associative" object.

In the object-oriented modelling [6 – 7] further classification of connections is given in comparison with previous theories. An object can be a part of an aggregation, a composition, an association or be inherited from another object.

It needs to be mentioned, that in all of the theories above connections are used rather technically to search for the corresponding elements, elements in the hierarchy, possessed and containing elements. The qualitative assessment of connections to optimize solutions of problems almost isn't conducted. All connections are treated equally. There is no extraction of essential in any way connections to optimize tasks being solved.

The goal of this paper is to create a classification of connections that can be further used to increase efficiency and effectiveness of control algorithms.

A typeof function. A subject domain (SD) is a tuple $(E_i(d_j), V_i(d_j), P_i(d_j))$, where E is a set of objects of an SD d_j, V is a set of connections of an SD d_j, and P is a set of mass problems of an SD d_j.A set of connections can be represented as

$$V = \{v_i\} = \{e^1_{v_i}, e^2_{v_i}\},$$

where $e^1_{v_i}$, $e^2_{v_i}$ – objects that are in the connection v_i.

A state s_i of an SD d is a tuple $< X_i, Y_i, Z_i >$, where X is a state of attributes of the SD d, Y is a state of connections of the SD d, Z_i is a set of actions that can be performed in the current state s_i.

Let us introduce a notion of an instance of a connection between instances of objects. An instance of a connection between instances of objects corresponds to the connection between corresponding objects. So if two objects are connected with a connection, two instances of those objects can be correspondingly connected with an instance of that connection. The main difference is that the connection between objects cannot appear or disappear because that will change the SD itself. On the other hand, instances of connections can appear and disappear with time. Thus, the set Y consists of instances of connections:

$$Y = \{\gamma_i\} = \{\varepsilon^1_{\gamma_i}, \varepsilon^2_{\gamma_i}\},$$

where γ_i is an instance of a connection $\varepsilon^1_{\gamma_i}$, $\varepsilon^2_{\gamma_i}$ are instances of objects that are in connection. Let us introduce a set of instances of objects in the current state s_i of an SD:

$$I(s_i) = I_i.$$

Instances of objects are considered as different if they have at least one attribute with different values. Let us denote Ω a set of all possible states of an SD. Let us denote Ψ a set of all possible sets of instances of objects of an SD:

$$\Psi = \{I(s) \mid \forall s \in \Omega\}.$$

Let us denote a set of all possible states of connections of an SD:

$$\Lambda = \{Y_i\}.$$

Let us introduce a function

$$typeof : I \cup Y \to E \cup V,$$

which matches an instance of an object or of a connection to a corresponding object or a connection.

Classification of connections. Connections between objects define structure of an SD and define its properties. They also define a set of all possible states of connections between instances of objects. Let us introduce a classification of connections between objects of an SD for its further detailed analysis.

Fraternal connections. There are connections between instances of objects which do not appear or disappear during functioning of an SD (that includes changes of a state of an SD and solution of mass problems). Those connections are set with the creation of an SD and define its structure during its lifetime. Those connections define main properties of that SD. Let us call those connections "fraternal connections". Note, that fraternal connections are not introduced especially to solve mass problems of an SD but only to maintain structure of objects of an SD. Corresponding instances of connections between instances of objects of an SD never change instances of objects on their ends.

Let us denote a set of fraternal connections as V^b, and a set of instances of fraternal connections as Y^b. Let us denote for each mass problem a set of fraternal connections that are used in its solution:

$$f^p_{vb} : P \to \{V^b_i \mid V^b_i \subset V^b\}.$$

Logical connections. During the creation of an SD with the purpose of solving mass problems in it connections between objects of this SD are introduced to allow instances of those objects to interact with each other for solving mass problems in that SD. Let us call those connections logical and denote the corresponding set V^l. At the same time, the same set of objects can describe different SDs with different sets of mass problems if those SDs have different sets of logical connections between those objects. We can describe the relation between fraternal and logical connection as following:

$$V^l = V \setminus V^b.$$

Also, let us define a function that will match each mass problem with a set of logical connections that are necessary for its solution:

$$f^p_{vl} : P \to \{V^l_i \mid V^l_i \subset V^l\}.$$

Stability and inevitability of a connection. Let us introduce two characteristics of connections of an SD: level of stability and a level of inevitability.

Let us define a level of stability of a connection as a probability that the corresponding instance of that connection at the next step will not disappear:

$$stability : V \to [0;1],$$

$$stability(v) = P(y \in Y_i \mid y \in Y_{i-1}),$$

where Y_{i-1}, Y_i are states of connections at $(i-1)-th$ and $i-th$ states of an SD correspondingly, $v = typeof(y)$. Therefore, the level of stability for fraternal connections is equal to one.

Let us define a level of inevitability of a connection as a probability that at the next step an instance of that connection will be created between instances of objects of an SD:

$$inevitability : V \to [0;1],$$

$$inevitability(v) = P(y \in Y_i \mid y \notin Y_{i-1}),$$

where Y_{i-1}, Y_i are states of connections at $(i-1)$-th and i -th states of an SD correspondingly, $v = typeof(y)$. Therefore, the level of inevitability of fraternal connections is equal to zero, because new fraternal connections are not created in an SD.

Let us note that the higher the level of stability a connection is, the more fundamental properties of an SD it defines. For example, a connection between a pupil and a teacher has a higher level of stability in an SD "school" than a connection between a pupil and a library, because the pupil can stop using the library, but it is highly unlikely that the pupil will stop speaking with the teacher. It is expected that a change in the most stable connections between instances of objects of an SD may cause the most fundamental and radical changes in the quality of a state of that SD. Also, let us notice that the most stable connections are aimed to solve long-term problems of an SD.

Connections with high stability and high inevitability as well as fraternal connections define fundamental properties of an SD for a long term. Nevertheless, apart from fraternal connections, their instances may actually disappear. Those connections define fundamental principles of functioning of an SD for solving most of its main problems in a long-term perspective.

Connections with high stability and low inevitability on the other hand allow solving long-term but rarely appearing problems. If we take a country as an example of an SD, then the instances of those kind of connections will be created when that country will enter a state of war. New connections that appear in these cases are stable and long-term, but appear rarely due to low frequency of necessity of solution of those kind of mass problems.

Connections with low stability and high inevitability appear in an SD to solve short-term frequent mass problems, for example transporting people with public transport (a connection "bus – person").

Connections with low stability and low inevitability appear in an SD to solve short-term infrequent mass problems, for example distinguishing of a fire at a factory by staff or local elections.

Connections with normal stability and normal inevitability appear in an SD to solve middle-term periodic problems.

To determine boundaries of low, normal and high values we can use mean value and standard deviation. Let μ be the mean value of stability (inevitability), σ – its standard deviation. Then let us consider low all connections with stability (inevitability) lower than $\mu - \delta\sigma$, and high – all connections with stability (inevitability) higher than $\mu + \delta\sigma$, where δ is a parameter which can be preset or chosen depending on the distribution type of stability (inevitability) to achieve the necessary percentage distribution of connections between those three groups. For example, for a normal distribution we can take $\delta = 1$ to get approximately 68 % of connections in the normal group, 16 % of connections in the low group and 16% of connections in the high group.

To calculate stability and inevitability functions let us consider a probability of a usage of an influence that can destroy or create an instance of that connection, and let us consider a probability that the influence will actually create or destroy an instance of that connection. More formally, let us denote $P(u)$ – a probability that an influence u will be used. Let us introduce two functions:

$$createprobability(v,u) = P(y \in Y_i \mid y \notin Y_{i-1}, u)$$

– a probability that a new instance y of a connection v will be created directly after usage of an influence u;

$$destroyprobability(v,u) = P(y \notin Y_i \mid y \in Y_{i-1}, u)$$

– a probability that an instance y of a connection v will be created directly after usage of an influence u.

Having defined that, let us notice the dependencies:

$$stability(v) = 1 - \sum_{u \in U_i} P(u) destroyprobability(v,u),$$

$$inevitability(v) = \sum_{u \in U_i} P(u) createprobability(v,u),$$

given that

$$\forall u \in U_i \sum_{u \in U_i} P(u) = 1.$$

For our convenience let us introduce a function

$$instability(v) = 1 - stability(v) =$$
$$= \sum_{u \in U_i} P(u) destroyprobability(u) \cdot$$

This function corresponds to a probability that an instance of the given connection will be destroyed at the next step.

Thus, calculation of functions *stability* and *inevitability* can be reduced to calculation of functions $P(u)$, $createprobability(v,u)$, $destroyprobability(v,u)$ for all possible influences for the current step. For calculation of those functions let us use a history of functioning of an SD. A history of functioning H of an SD is a sequence $H = \{(s_i, u_i)\}_{i=1}^{n}$ where $s_i \in \Omega$ is a state of the SD at the i-th step, is an influence which was executed at the i-th step, which transferred the SD to the new state. Functions, mentioned above, are suggested to be calculated as follows:

$$P(u) = \frac{1}{n} \sum_{u_i = u} 1,$$

$$createprobability(v,u) =$$
$$= \left(\sum_{u_i = u} \left(\left(\sum_{\substack{y \in Y_{i+1} \\ v = typeof(y) \\ y \in Y_{i+1} \wedge y \notin Y_i}} 1 \right) \left(\sum_{\substack{y \in Y_{i+1} \\ v = typeof(y)}} 1 \right)^{-1} \right) \right) \left(\sum_{u_i = u} 1 \right)^{-1}$$

$$destroyprobability(v,u) =$$
$$= \left(\sum_{u_i = u} \left(\left(\sum_{\substack{y \in Y_i \\ v = typeof(y) \\ y \notin Y_{i+1} \wedge y \in Y_i}} 1 \right) \left(\sum_{\substack{y \in Y_i \\ v = typeof(y)}} 1 \right)^{-1} \right) \right) \left(\sum_{u_i = u} 1 \right)^{-1}.$$

Let us note that the same influence in the same state can lead both to creation and destruction of instances of the same connection.

Given functions allow building a Markov's process [8] or a stochastic Petri net [9] to model a process of creation and destruction of connections in the given SD.

It is also possible to extract another group of connections which objects belong to the set of defining objects [10]. Those connections may define a subject domain and give essential information about it as defining attributes [10].

Example. Let us consider a relational database. The relations are objects, tuples are instances of objects, and stored procedures are influences. It is possible to calculate introduced characteristics using the transaction history as a history of changes of the state of that subject domain. Using that information it is possible to predict and estimate the changes in the connections between tuples of objects, find stable and unstable clusters of objects in the model. This can be used to split the storage of information between a relational database for the unstable part of the data model, and a NoSQL database for the stable part of the data model to increase performance of the system.

Conclusion. Introduced results can be used for predicting creation and destruction of connections between instances of objects, probability analysis of transitions between states of an SD in perspective of its connections, measurement of stability of an SD, and comparison of SDs in regard to their stability.

Further research can be aimed for developing algorithms of efficient classification of connections, and developing further classification of connections. It will permit to develop control algorithms that will use this classification to increase effectiveness and efficiency of management decisions.

References

1. Malakhov E.V. Rasshireniie operatsiy nad metamodelyami predmetnih oblastey s uchetom massovyh problem. [Expansion of Operations on Subject Domains Metamodels Taking into Account the Mass Problems], (2010), *Eastern-European Journal of Enterprise Technologies*, Kharkovm Ukraine, *Technology Center*, Vol. 5/2 (47), pp. 20 – 24 (In Russian).

2. Chartrand G., and Zhang P., (2012), A First Course in Graph Theory (Dover Books on Mathematics). New York: *Dover Publications,* 464 p.

3. Jungnickel D., (2012), Graphs, Networks and Algorithms (4th ed.), Germany: Springer, 677 p.

4. Connolly T.M., and Begg C.E., (2001), Database Systems: a Practical Approach to Design, Implementation, and Management (3rd ed.). Boston: Addison-Wesley. 1312 p.

5. Kimball R., and Ross M., (2013), The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling (3rd ed.), *Indianapolis*, Indiana: Wiley. 600 p.

6. Meyer B., (1997), Object-Oriented Software Construction (2 nd ed.). New Jersey: Prentice hall PTR. 1296 p.

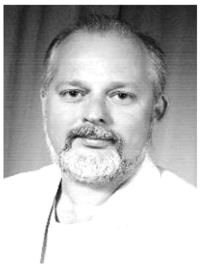
7. Fowler M., (2003), UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition), *Addison-Wesley Professional,* Boston, 208 p.

8. Stroock D.W., (2005), An Introduction to Markov Processes (Graduate Texts in Mathematics). Germany: Springer, 178 p.

9. Haas P.J., (2002), Stochastic Petri Nets: Modelling, Stability, Simulation (Springer Series in Operations Research and Financial Engineering), Germany, Springer, 510 p.

10. Mezhuyev V., Malakhov E., and Shchelkonogov D., (2015), The Method and Algorithms to Find Essential Attributes and Objects of Subject Domains [text] *IEEE 2015 International Conference on Computer, Communication and Control Technology (I4CT 2015)*, Kuching, Malaysia, pp. 310 – 314.

Received 30.05.2015



Malakhov
Eugene V., ScD. (Tech.),
Professor, Head of Dept. of Mathematical Support of Computer Systems, Odessa I. I. Mechnikov National University.
E-mail:
eugene.malakhov@onu.edu.ua

Shchelkonogov
Denys O.,
Master of Applied Mathematics, graduate student of the Dept. Of Mathematical Support of Computer Systems, Odessa I.I.Mechnikov National University.
E-mail: delis91@gmail.com